Timothy Zwart

Assignment 1

**Date Assigned: 08/31/2018**

**Due: Midnight 09/07/2018 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Consider the following problem:

**Input:** an array, A, of n sorted integers (positive, negative, or 0) that

A[0] <= A[1] <= A[2] <=…A[n-2] <= A[n-1]

**Output:** re-arrange elements in A such that:

Element at even position (i.e., A[0], A[2]) are less than or equal to both of its neighbors

Element at odd position (i.e., A[1], A[3]) are greater than or equal to both of its neighbors

A[0] <= A[1] >= A[2] <= A[3] >= A[4] <= A[5]…

Design an algorithm that solves this problem.

**(i) describe the idea behind your algorithm in English (5 points);**

**(ii) provide pseudocode (10 points);**

English: Iterate a loop by two spots starting from the first odd numbered element. Set current element value to a new variable. Switch current element value to next element value in the array. Then set the next element equal to the variable value.

Pseudocode: For (i=1; I < length – 1; i+2){

Temporary value = A[i];

A[i] = A[i+1];

A[i+1]= temporary value;

}

1. Consider the following problem:

**Input:** an array A of n integers (positive, negative, or 0), elements sorted in ascending order.

**Output:** if there exists a majority element.

An element is a majority if it appears more than n/2 times. For example, if the input list is:

{0, 0, 0, 0, 0, 0, 1, 1, 2, 4, 7}

The output should be 0, as 0 appears 6 times (>n/2 = 11/2 times).

However, if the input list is:

{0, 0, 0, 1, 1, 2, 3, 10, 10}

The majority element does not exist.

Design an algorithm that solves this problem.

**(i) describe the idea behind your algorithm in English (5 points);**

**(ii) provide pseudocode (10 points);**

English: Iterate through the array. Provide a counter and a variable for current element value. Each time the number appears count. And when not the number appears count down. If counter reaches 0 make current element being checked in the array into the current element variable. Upon ending the array run another loop where counter starts at zero and the current element variable is checked with each element in the array and if it is equal add 1 to the counter. If the Counter is greater than length/2, then there is a majority element. If not then there is no majority element.

Pseudocode: Counter=0;

Major number =A[0];

for(i=0;I < A.length;i++){

if(major number ==A[i]){

count++;

}else{

Count- -

If(count==0){

Major number=A[i];

}

}

}

Count =0

For(i=0; i<length; i++){

If(major number== A[i]){

Count++

}

}

If(count > A.length/2){

Print Major does exist

}else{print major does not exist}